

10.2-pandas

December 19, 2024

Pandas-DataFrame And Series Pandas is a powerful data manipulation library in Python, widely used for data analysis and data cleaning. It provides two primary data structures: Series and DataFrame. A Series is a one-dimensional array-like object, while a DataFrame is a two-dimensional, size-mutable, and potentially heterogeneous tabular data structure with labeled axes (rows and columns).

```
[1]: import pandas as pd
```

```
[3]: ## Series  
##A Pandas Series is a one-dimensional array-like object that can hold any data  
↳type. It is similar to a column in a table.  
  
import pandas as pd  
data=[1,2,3,4,5]  
series=pd.Series(data)  
print("Series \n",series)  
print(type(series))
```

```
Series  
0    1  
1    2  
2    3  
3    4  
4    5  
dtype: int64  
<class 'pandas.core.series.Series'>
```

```
[4]: ## Create a Series from dictionary  
data={'a':1,'b':2,'c':3}  
series_dict=pd.Series(data)  
print(series_dict)
```

```
a    1  
b    2  
c    3  
dtype: int64
```

```
[5]: data=[10,20,30]  
index=['a','b','c']
```

```
pd.Series(data,index=index)
```

```
[5]: a    10  
     b    20  
     c    30  
     dtype: int64
```

```
[15]: ## Dataframe  
## create a Dataframe from a dictionary of list  
  
data={  
    'Name': ['Mukesh', 'John', 'Jack'],  
    'Age': [25,30,45],  
    'City': ['Bangalore', 'New York', 'Florida']  
}  
df=pd.DataFrame(data)  
print(df)  
print(type(df))
```

```
      Name  Age    City  
0  Mukesh   25  Bangalore  
1   John   30   New York  
2   Jack   45   Florida  
<class 'pandas.core.frame.DataFrame'>
```

```
[16]: ## Create a Data frame From a List of Dictionaries  
  
data=[  
    {'Name': 'Mukesh', 'Age': 32, 'City': 'Bangalore'},  
    {'Name': 'John', 'Age': 34, 'City': 'Bangalore'},  
    {'Name': 'Bappy', 'Age': 32, 'City': 'Bangalore'},  
    {'Name': 'JACK', 'Age': 32, 'City': 'Bangalore'}  
]  
df=pd.DataFrame(data)  
print(df)  
print(type(df))
```

```
      Name  Age    City  
0  Mukesh   32  Bangalore  
1   John   34  Bangalore  
2   Bappy   32  Bangalore  
3   JACK   32  Bangalore  
<class 'pandas.core.frame.DataFrame'>
```

```
[17]: df1=pd.read_csv('sales_data.csv')  
      df1.head(5)
```

```
[17]: Transaction ID      Date Product Category      Product Name \
0      10001  2024-01-01      Electronics      iPhone 14 Pro
1      10002  2024-01-02  Home Appliances      Dyson V11 Vacuum
2      10003  2024-01-03      Clothing      Levi's 501 Jeans
3      10004  2024-01-04      Books      The Da Vinci Code
4      10005  2024-01-05  Beauty Products  Neutrogena Skincare Set
```

```
      Units Sold  Unit Price  Total Revenue      Region Payment Method
0           2      999.99      1999.98  North America      Credit Card
1           1      499.99      499.99      Europe      PayPal
2           3       69.99      209.97      Asia      Debit Card
3           4       15.99       63.96  North America      Credit Card
4           1       89.99       89.99      Europe      PayPal
```

```
[13]: df1.tail(5)
```

```
[13]: Transaction ID      Date Product Category \
235      10236  2024-08-23  Home Appliances
236      10237  2024-08-24      Clothing
237      10238  2024-08-25      Books
238      10239  2024-08-26  Beauty Products
239      10240  2024-08-27      Sports
```

```
      Product Name  Units Sold  Unit Price \
235 Nespresso Vertuo Next Coffee and Espresso Maker      1      159.99
236      Nike Air Force 1 Sneakers      3       90.00
237 The Handmaid's Tale by Margaret Atwood      3       10.99
238 Sunday Riley Luna Sleeping Night Oil      1       55.00
239 Yeti Rambler 20 oz Tumbler      2       29.99
```

```
      Total Revenue      Region Payment Method
235      159.99      Europe      PayPal
236      270.00      Asia      Debit Card
237      32.97  North America      Credit Card
238      55.00      Europe      PayPal
239      59.98      Asia      Credit Card
```

```
[18]: ### Accessing Data From Dataframe
df
```

```
[18]: Name  Age  City
0  Mukesh  32  Bangalore
1   John  34  Bangalore
2  Bappy  32  Bangalore
3   Jack  32  Bangalore
```

```
[19]: df['Name']
```

```
[19]: 0    Mukesh
      1     John
      2    Bappy
      3     Jack
      Name: Name, dtype: object
```

```
[20]: df.loc[0]
```

```
[20]: Name    Mukesh
      Age      32
      City  Bangalore
      Name: 0, dtype: object
```

```
[21]: df.iloc[0]
```

```
[21]: Name    Mukesh
      Age      32
      City  Bangalore
      Name: 0, dtype: object
```

```
[22]: df
```

```
[22]:   Name  Age  City
0  Mukesh  32  Bangalore
1   John  34  Bangalore
2  Bappy  32  Bangalore
3   Jack  32  Bangalore
```

```
[23]: ## Accessing a specified element
      df.at[2, 'Age']
```

```
[23]: 32
```

```
[24]: df.at[2, 'Name']
```

```
[24]: 'Bappy'
```

```
[25]: ## Accessing a specified element using iat
      df.iat[2,2]
```

```
[25]: 'Bangalore'
```

```
[26]: df
```

```
[26]:   Name  Age  City
0  Mukesh  32  Bangalore
1   John  34  Bangalore
2  Bappy  32  Bangalore
```

```
3    Jack    32    Bangalore
```

```
[28]: ### Data Manipulation with Dataframe  
df
```

```
[28]:      Name  Age      City  
0  Mukesh   32  Bangalore  
1   John   34  Bangalore  
2  Bappy   32  Bangalore  
3   Jack   32  Bangalore
```

```
[31]: ## Adding a column  
df['Salary']=[50000,60000,70000, 80000]  
df
```

```
[31]:      Name  Age      City  Salary  
0  Mukesh   32  Bangalore   50000  
1   John   34  Bangalore   60000  
2  Bappy   32  Bangalore   70000  
3   Jack   32  Bangalore   80000
```

```
[32]: ## Remove a column  
df.drop('Salary',axis=1,inplace=True)
```

```
[33]: df
```

```
[33]:      Name  Age      City  
0  Mukesh   32  Bangalore  
1   John   34  Bangalore  
2  Bappy   32  Bangalore  
3   Jack   32  Bangalore
```

```
[34]: ## Add age to the column  
df['Age']=df['Age']+1  
df
```

```
[34]:      Name  Age      City  
0  Mukesh   33  Bangalore  
1   John   35  Bangalore  
2  Bappy   33  Bangalore  
3   Jack   33  Bangalore
```

```
[35]: df.drop(0,inplace=True)
```

```
[36]: df
```

```
[36]:      Name  Age      City  
1   John   35  Bangalore
```

```
2 Bappy 33 Bangalore
3 Jack 33 Bangalore
```

```
[37]: df=pd.read_csv('sales_data.csv')
df.head(5)
```

```
[37]:
```

	Transaction ID	Date	Product Category	Product Name \
0	10001	2024-01-01	Electronics	iPhone 14 Pro
1	10002	2024-01-02	Home Appliances	Dyson V11 Vacuum
2	10003	2024-01-03	Clothing	Levi's 501 Jeans
3	10004	2024-01-04	Books	The Da Vinci Code
4	10005	2024-01-05	Beauty Products	Neutrogena Skincare Set

	Units Sold	Unit Price	Total Revenue	Region	Payment Method
0	2	999.99	1999.98	North America	Credit Card
1	1	499.99	499.99	Europe	PayPal
2	3	69.99	209.97	Asia	Debit Card
3	4	15.99	63.96	North America	Credit Card
4	1	89.99	89.99	Europe	PayPal

```
[39]: # Display the data types of each column
print("Data types:\n", df.dtypes)
```

Data types:

```
Transaction ID      int64
Date                object
Product Category    object
Product Name        object
Units Sold          int64
Unit Price          float64
Total Revenue       float64
Region              object
Payment Method      object
dtype: object
```

```
[40]: # Describe the DataFrame
print("Statistical summary:\n", df.describe())
```

Statistical summary:

	Transaction ID	Units Sold	Unit Price	Total Revenue
count	240.00000	240.000000	240.000000	240.000000
mean	10120.50000	2.158333	236.395583	335.699375
std	69.42622	1.322454	429.446695	485.804469
min	10001.00000	1.000000	6.500000	6.500000
25%	10060.75000	1.000000	29.500000	62.965000
50%	10120.50000	2.000000	89.990000	179.970000
75%	10180.25000	3.000000	249.990000	399.225000
max	10240.00000	10.000000	3899.990000	3899.990000

```
[41]: df.describe()
```

```
[41]:
```

	Transaction ID	Units Sold	Unit Price	Total Revenue
count	240.00000	240.000000	240.000000	240.000000
mean	10120.50000	2.158333	236.395583	335.699375
std	69.42622	1.322454	429.446695	485.804469
min	10001.00000	1.000000	6.500000	6.500000
25%	10060.75000	1.000000	29.500000	62.965000
50%	10120.50000	2.000000	89.990000	179.970000
75%	10180.25000	3.000000	249.990000	399.225000
max	10240.00000	10.000000	3899.990000	3899.990000

```
[ ]:
```